www.elsevier.com/locate/yjtbi

Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics

L. Tournier, M. Chavesa

PII: S0022-5193(09)00265-3
DOI: doi:10.1016/j.jtbi.2009.06.006
Reference: YJTBI 5595

To appear in: *Journal of Theoretical Biology*

Received date: 9 October 2008
Revised date: 5 June 2009
Accepted date: 5 June 2009

Cite this article as: L. Tournier and M. Chavesa, Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics, *Journal of Theoretical Biology*, doi:10.1016/j.jtbi.2009.06.006

Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics

L. Tournier[*,a,1], M. Chaves[a]

[a] *INRIA Sophia, COMORE Team, 2004, Route des lucioles, B.P. 93, F-06902 Sophia-Antipolis Cedex, France*

Abstract

Biological networks of large dimensions, with their diagram of interactions, are often well represented by a Boolean model with a family of logical rules. The state space of a Boolean model is finite, and its asynchronous dynamics are fully described by a transition graph in the state space. In this context, a model reduction method will be developed for identifying the active or operational interactions responsible for a given dynamic behaviour. The first step in this procedure is the decomposition of the asynchronous transition graph into its strongly connected components, to obtain a "reduced" and hierarchically organized graph of transitions. The second step consists of the identification of a partial graph of interactions and a sub-family of logical rules that remain operational in a given region of the state space. This model reduction method and its usefulness are illustrated by an application to a model of programmed cell death. The method identifies two mechanisms used by the cell to respond to death-receptor stimulation and decide between the survival or apoptotic pathways.

Key words: Boolean networks, Asynchronous transition graph, Model reduction, Apoptosis, NFκB signalling pathway

[*]Corresponding author.

Email addresses: laurent.tournier@curie.fr (L. Tournier), madalena.chaves@sophia.inria.fr (M. Chaves)

[1]Present address: Institut Curie, INSERM-U900, Mines ParisTech, 26, rue d'Ulm, F-75005 Paris, France.

1. Introduction

Discrete and, in particular, Boolean models have been playing an increasingly important role in the study and analysis of complex biological systems [27, 25, 5, 26, 11, 6] (note that even though [25] deals with piecewise linear differential models, a large part of the analysis of such systems pertains to the discrete framework). Based on the idea that each variable may take values only on a finite set, discrete models offer a very attractive framework for the systematic study of the dynamics of large systems, which may range from a few to hundreds of variables and their interactions.

The discrete modelling approach is highly relevant for many of the currently data acquisition techniques for signalling and genetic regulatory networks (microarrays, fluorescence markers, electrophoretic mobility shift assays, etc.) which involve more qualitative measurements. A discrete system may be expected to give a good idea of the system's dynamics from the available data (for example, on multistationary, stability, or oscillatory behavior). At the same time, Boolean networks provide a measure of the robustness of a system [5], since this dynamical information is essentially independent of the system's parameters (such as kinetic rates, binding rates, or degradation constants) and depends only on the interconnection structure, and the logical function characterizing each node. For instance, the transition graph of the network indicates how much a given trajectory may be affected by perturbations, or whether the system is capable of maintaining a given dynamical behaviour despite fluctuations in the environment. Indeed, a major advantage of discrete and Boolean modelling is the possibility, for systems of reasonable size, of fully characterizing all *qualitative* dynamical trajectories of a particular network, based simply on the structure of links and interactions between nodes. This general characterization and "easier" handling of the state space, counterbalance the loss of detailed information on time evolution and (more realistic) continuous concentration changes.

The study of complex systems with many variables frequently raises questions concerning the possibility of simplifying or reducing the system in some way. To simplify the analysis and gain intuition, it is often useful to identify a smaller, easier to analyze, family of variables and interactions that still faithfully describe the original system and exhibit the same overall qualitative dynamics. Likewise, it is often of interest to find out whether different groups of variables are associated with different dynamics [29]. Another related question is whether all interactions operate at all times, or whether different groups of interactions become active or operational at different times, in response to a precise context [18]. Similarly, find-

ing interactions which prevent a given target function is useful from the point of view of therapeutical interventions, for instance [19, 26]. These are all challenging problems, and while some model reduction methods exist, they are generally aimed at special classes of systems (a survey of methods used for control systems can be found in [1]; a method for identifying the variables responsible for complex cell behavior was proposed in [29]). One of the objectives of this paper is to show that, to some extent, answers to these questions may be obtained through the discrete systems framework and some of its techniques.

In this context, a model reduction technique is proposed in this paper, for the analysis of asynchronous Boolean networks motivated by biological (namely, signalling or genetic) regulatory networks of intermediate dimension (*e.g.*, 8-20 variables). The model reduction technique combines and adapts two methods: the classical decomposition of a graph into its strongly connected components [7] and an identification algorithm described in [20, 36]. The first part of the model reduction technique involves the simplification of the asynchronous transition graph into its strongly connected components. These components are then organized into hierarchical levels, such that any given trajectory can only move into the next level in the hierarchy, but never into the previous level. A new "reduced" transition graph is then constructed which describes the transitions between the strongly connected components (Section 3). The second part of the model reduction involves the identification of the "operational" network (active interactions) that is responsible for the dynamics of the system from a given level in the hierarchy (Section 4).

Finally, in Section 5, biological knowledge is incorporated into the Boolean model by associating a (fixed) matrix of transition probabilities to the graph edges, a process which corresponds to generating a Markov chain. Several relevant quantities can then be computed, such as the expected times for convergence to a given attractor.

The methods proposed above are illustrated by an application to an apoptosis (or programmed cell death) network [28, 6], with $n = 12$, as described in Section 2.3. The dynamics of the network in response to death-receptor stimulation is studied, and two core groups of variables and pathways are identified: it is shown that these correspond to two mechanisms responsible for the decision between programmed cell death or cell survival. In addition, associating a transition probabilities' matrix to the apoptosis network allows us to estimate, among other quantities, the probability of cell survival or death upon stimulation of death receptors.

2. Asynchronous Boolean models of gene regulatory networks

Discrete dynamical systems have been widely studied for decades, as they provide a good mathematical and algorithmic framework to model systems where variables are known or assumed to take values only in a finite set (as opposed to a continuum of values). In particular, the discrete framework has been often applied to model biological (for instance, genetic) regulatory networks [17, 33]. A mathematical discrete model consists of a finite set of variables together with a family of *activation functions*, which describe the interactions among the variables. Both variables and activation functions are only allowed to take values in a finite set. A common example is that of Boolean variables, which take only the values 1 or 0 (ON or OFF). The interactions among variables can be described by a (finite) directed graph, called *interaction graph*. This graph, together with the family of activation functions, define the structure of a discrete system. Each variable will evolve according to a given rule, constructed from the interaction graph. In order to describe the dynamics of such a system, a fixed time unit is postulated, and the system is updated at integer multiples of this time unit (between two updates, the system's variables are assumed to be constant). To update the system one must define a strategy that determines the order in which the variables are evaluated over time.

Two updating strategies have been studied in the literature. The first one is the *synchronous* strategy, where all variables are simultaneously updated at each discrete instant (see [17] for an extensive study of this synchronous strategy; see also [30, 37]). The dynamics resulting from the synchronous strategy presents some nice mathematical properties (mainly, the transition graph is deterministic) that allow one to simulate high-dimensional networks, in order to find statistically relevant types of dynamical behavior [17]. However, if one wants to model a given biological system in a more realistic manner, the synchronous updating strategy is quite a strong assumption, implying that all variables are produced or degraded at the same rates. Thus other approaches have been proposed, by developing *asynchronous* strategies, where the discrete variables are updated in a heterogeneous way over time. For example, the variables may be updated in a randomly generated order, or according to an order based on known properties (for instance, transcription or translation are slower processes than protein-protein binding) [5]. Discrete networks with asynchronous updating orders are often called Thomas' networks [33, 35, 34], and are much better suited to model the dynamical behavior of biological regulatory networks. Before giving more details on the type of asynchronicity that will be used in this paper, we first recall some basic definitions

about the structure of discrete networks.

2.1. Structure of a Boolean network

In this paper, we will consider *Boolean* networks, where the variables (which represent, for instance, the level of expression of genes, or the level of concentration of different species, such as proteins) can take only two qualitative values. "0" represents a basal level (inhibition -or weak activation- of the transcription of a gene, or *absence*[2] of a biochemical species) and "1" represents a high level (activation of the transcription of a gene, or *presence* of a species). We note here that there exist more general frameworks, where the variables can take more than two qualitative values (see for instance [27, 4] or [31]).

As most of the work on discrete or Boolean gene networks is based on the same mathematical objects (with slightly different definitions), the following part is only a brief summary of the main definitions and notations that will be used in the rest of this paper (for a detailed explanation of these definitions, one can refer to the extensive literature on discrete networks). Let us begin with the definition of the interaction graph, which is the core of the structure of a discrete model.

Definition 1 (Interaction Graph) *The interaction graph of a n-dimensional Boolean network is defined by* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $\mathcal{V} = \{v_1, \ldots, v_n\}$ *is the set of nodes (each node may represent a biological species) and* $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ *is the set of directed edges (representing the interactions between these species). The edge* (v_j, v_i) *exists if node* v_j *influences node* v_i *(e.g.* v_j *activates or inhibits* v_i*).*

Each node $v_i \in \mathcal{V}$ has a set of inputs (possibly empty), which are the nodes that influence its evolution:

$$\mathcal{I}(v_i) = \left\{ v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E} \right\} \subset \mathcal{V}.$$

For each $v_i \in \mathcal{V}$, the cardinality of the set $\mathcal{I}(v_i)$ (the number of its inputs) is often called the *connectivity* of node v_i and is generally denoted by k_i. In order to give a complete definition of the structure of the network, we now define the activation functions.

[2] generally, a given species is considered to be *absent* if its concentration is lower than a given threshold, *present* otherwise

Definition 2 *The structure of a n-dimensional Boolean network is defined by an interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ together with a collection $\mathcal{F} = \{f_i : \ i = 1, \ldots, n\}$, of Boolean functions:*

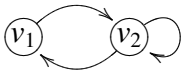
$$f_i : \{0,1\}^{k_i} \longrightarrow \{0,1\},$$

where, for each $i \in \{1, \ldots, n\}$, f_i designates the activation function of node v_i, and k_i its number of inputs.

Let x_i denote the Boolean variable associated with node v_i. The updated value of x_i, denoted by x_i' is therefore given by:

$$x_i' = f_i\left(x_{i_1}, \ldots, x_{i_{k_i}}\right), \quad \text{where: } \left\{v_{i_1}, \ldots, v_{i_{k_i}}\right\} = \mathcal{I}(v_i).$$

To illustrate these definitions, let us consider a simple 2-dimensional network, given by the following interaction graph and set of rules:



$$\begin{cases} f_1(x_2) & = & \overline{x_2}, \\ f_2(x_1, x_2) & = & \text{xor}(x_1, x_2) = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}). \end{cases}$$

The notations used in this example are classical in Boole's algebra: if x and y denote two Boolean variables, $\overline{x}$ denotes the negation of x, and $x \wedge y$, $x \vee y$ denote, respectively, the product (logical function *and*) and summation (logical function *or*) of x and y. The symbol xor denotes the exclusive *or*.

Remark 1 *The three following points emphasize modeling choices, induced by these definitions, that will be made in the rest of the paper.*

- *Multiple edges are not allowed in the interaction graph. If a particular element of the network v_i has two distinct influences on another element v_j, they will be represented by a unique edge in $\mathcal{G}$. Nevertheless, these two influences will appear in the logical rule of f_j.*
- *In the study of discrete models of biochemical networks, the arrows of the interaction graph are usually signed, indicating whether the arrow represents an* activation *($\overset{+}{\rightarrow}$ or $\rightarrow$) or an* inhibition *($\overset{-}{\rightarrow}$ or $\dashv$). It is to be noted that, for a general network given by Def. 2, it is not always possible to associate a sign to each arrow in an unequivocal manner. In the previous example, for instance, the arrows (v_1, v_2) and (v_2, v_2) cannot be signed as the interactions they represent are neither activations nor inhibitions. Nevertheless, in a Boolean network constructed from the description of a particular biological system, most of the interactions can be signed unequivocally. In the rest*

of the paper, activations and inhibitions will thus be labelled (respectively with '+' and '-') while all other interactions will stay unlabelled.

- *In the above approach, connectivity is defined prior to the activation functions. This can lead to inaccuracies if the real connectivity (as defined in [36]) of the function does not match its apparent connectivity. Thus, the function* $f(x_1, x_2) = (x_1 \wedge x_2) \vee (x_1 \wedge \overline{x_2})$ *has an apparent connectivity of* 2*, whereas its real connectivity is* 1 *(indeed* $f(x_1, x_2) = x_1$*). In the terminology of [30],* x_2 *is called a fictitious (or non essential) variable for function* f*. This issue becomes particularly important if one wants to identify a network from given data. It will be addressed in more details in Section 4.*

2.2. Synchronous vs asynchronous dynamics

Consider a n-dimensional network given by $\mathcal{N} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ (see Def. 2). The state space of $\mathcal{N}$ is the set $\Omega = \{0, 1\}^n$ whose cardinality is 2^n. As the state space is finite, one can represent the discrete dynamical behavior of the network with a finite directed graph, called *transition graph*. In order to define it properly, we need to assign an updating strategy for the network $\mathcal{N}$. Two main approaches have been considered in the literature. The first one is to consider a synchronous update of all nodes at each time (see [17]). If the state of the network at time t is given by the Boolean vector:

$$X(t) = (x_1(t), \ldots, x_n(t)) \in \Omega,$$

then the next state $X(t + 1)$ (also called the synchronous successor) is simply:

$$X(t + 1) = (x_1(t + 1), \ldots, x_n(t + 1)) \in \Omega,$$

where, for all i, $x_i(t + 1) = x_i'(t)$ (the updated value of $x_i(t)$). In this case, the temporal evolution of the network is *autonomous*, in the sense that any vector $X \in \Omega$ has a (unique) successor $F(X) = (f_1(X), \ldots, f_n(X))$, and that successor is independent of time t. We can then construct a directed transition graph: its set of nodes is Ω and its set of directed edges is defined by the "successor" function. The main property of the synchronous graph is that it is *deterministic*, *i.e.* each state has a unique successor. In particular, this property implies that each connected component of the graph contains a unique attractor, and this attractor is either a cycle or a fixed point. More precisely, the connected components are in fact the basins of attraction of their attractor. The synchronous updating strategy is a very strong assumption, not very realistic if one wants to model the dynamical behavior of a given biological system.

Actually, as discrete interactions are coarse-grained models of sometimes very complex biochemical processes (often implying several biochemical reactions), it is preferable to consider time-dependent, asynchronous updating strategies (historically introduced by R. Thomas [33]). In order to give a precise definition of an asynchronous transition graph, we first introduce the following notation:

- For each $X \in \Omega = \{0,1\}^n$, $F(X) \in \Omega$ designates the synchronous successor of X:
$$\forall i \in \{1,\ldots,n\},\ F_i(X) = x'_i.$$
- For each $X \in \Omega$ and each $i \in \{1,\ldots,n\}$, $\widetilde{X}^i$ designates the vector:
$$\widetilde{X}^i = (x_1,\ldots,x_{i-1},\overline{x_i},x_{i+1},\ldots,x_n) \in \Omega.$$
- For each $X \in \Omega$, let $U(X) = \left\{v_i \in \mathcal{V} \mid x_i \neq x'_i\right\} \subset \mathcal{V}$. $U(X)$ designates the (possibly empty) set of nodes that can actually be updated when the system is in the state X.
- An element $X^* \in \Omega$ is a steady state of the Boolean model if it satisfies: $U(X^*) = \emptyset$, that is $x^*_i = F_i(X^*)$, for all $i \in \{1,\ldots,n\}$.

We also state the following assumptions, that we will suppose verified throughout this paper:

Assumption 1 *At each discrete time t, at most one node is updated (no update means the network is in a steady state).*

Assumption 2 *Each state $X \in \Omega$ such that $X \neq F(X)$ has exactly $|U(X)|$ successors.*

The first assumption forbids the simultaneous update of several nodes (which is reasonable from the biological point of view), whereas the second one implies that every possible update is taken into account (*i.e.* if at state X the node v_i is liable to change, then that update *must* be present in the transition graph). With these assumptions, we can now define the (asynchronous) transition graph:

Definition 3 (Asynchronous Transition Graph) *The asynchronous transition graph of the network $\mathcal{N} = (\mathcal{V},\mathcal{E},\mathcal{F})$ is the directed graph $G = (V,E)$ where the set of nodes V is the state space $\Omega = \{0,1\}^n$ and the set of directed edges E is defined by:*
$$E = \left\{\left(X \to \widetilde{X}^i\right) \mid X \in \Omega, v_i \in U(X)\right\}.$$

In the comparison between synchronous and asynchronous dynamics, a well-known result is that, provided Assumptions 1 and 2 are satisfied, the asynchronous and synchronous steady states are the same. The proof follows immediately from the definition of steady state.

The non determinism of the asynchronous transition graph is a fundamental property. It allows to consider any possible trajectory implied by the structure of the network. If one wants to study one particular trajectory, then a particular path in the graph has to be chosen, which is equivalent to the choice of a particular updating strategy. Considering biological applications, such a choice will be based on the information available on the system. Unfortunately, the knowledge of the system is often incomplete. An advantage of this framework is the possibility to test and analyze different plausible *sets* of updating orders, as will be done later in this paper (it corresponds to the notion of *priority classes* developed in [11]). The main advantage of the asynchronous graph is that it comprises all the possible choices in a finite structure, which allows to find general dynamical properties valid whatever the updating strategy. Obviously, although finite, the size of the transition graph grows exponentially with the dimension of the system (in the Boolean case, its size is exactly 2^n). This limits the use of general graph algorithms to relatively low dimensional systems (on the order of n = 10-20), with respect to the synchronous case, where the dimension of the system under study can be higher [37].

2.3. Working example: an apoptosis signalling pathway

The model reduction method will be illustrated by application to an apoptosis network (Fig. 1). Apoptosis, or programmed cell death, is a physiological process which allows an organism to remove damaged or unwanted cells in a "clean" and natural way. The signalling pathways leading to apoptosis play fundamental roles in embryonic development and in adult organisms, by maintaining normal cellular homeostasis in organs and other cellular tissues [8]. Malfunctioning apoptotic pathways may lead to various diseases, such as cancer (in this case cells do not die, there is insufficient apoptosis), or immunodeficiency and infertility (in this case too many cells die, there is too much apoptosis) [8].

The apoptosis signalling pathway to be considered in this paper (Fig. 1) is based on the model presented in [6], which is, in fact, a discrete version of a continuous model of apoptosis first developed in [28]. A brief description of the network is provided next, and the reader is referred to [6, 28] and references therein for more details.

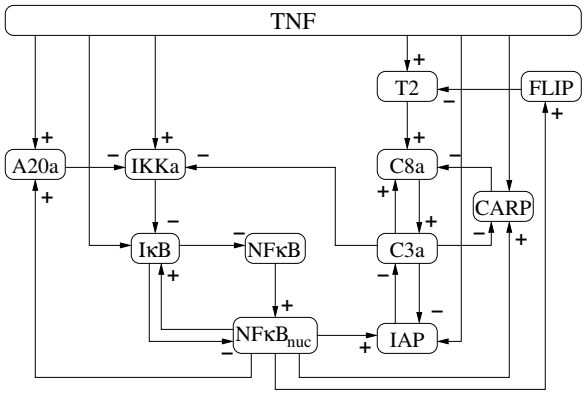


Figure 1: Interaction graph of the simplified model of regulation of apoptosis via the NFκB pathway. As noted in Remark 1, some edges do not have a fixed sign (influence of TNF on IκB, IAP and CARP).

The network is composed essentially of a pro-apoptotic and an anti-apoptotic pathway, which are activated by the same signal: stimulation of death receptors by a factor such as Tumor Necrosis Factor α (denoted TNF in Fig. 1). The pro-apoptotic pathway is based on the model developed in [9], and consists of a family of proteins called caspases, represented by active caspases 3 and 8 (resp., C3a and C8a) in Fig. 1. The caspases play the main role in apoptosis, as they cleave (or break into small pieces) the principal proteins in the cell, eventually leading to a "clean disposal" of the cell in response to an apoptotic signal. The anti-apoptotic pathway is based on the pioneering work of [16] and on the models developed in [16, 21], and represents the Nuclear Factor κB (NFκB) signalling pathway. The links among the two pathways are as yet, not fully characterized. It is well known that the NFκB signalling pathway is responsible for activating transcription of both pro- and anti-apoptotic genes [22], and thus plays an important role in regulating apoptosis. There is also evidence to show that caspase 3 has an inhibitory influence on inhibitor of IkB kinases (IKK) [12]. The components of the NFκB pathway are as follows (in biological terminology): Nuclear Factor κB in the cytoplasm (NFκB) and in the nucleus (NFκB$_{nuc}$); inhibitor of NFκB (IκB); inhibitor of IκB kinases (IKKa); inhibitor of apoptosis proteins (IAP); caspase-8 and -10-associated RING proteins (CARP); a protein associated with inhibition of complex T2 (FLIP); and a protein regulating IKK activity (A20a).

Binding of TNF to a death receptor activates the anti-apoptotic pathway and, after a certain delay (upon formation of a second complex, denoted by T2), the

pro-apoptotic pathway is also activated. The anti-apoptotic pathway activates synthesis of various proteins (IAP, CARP, FLIP) that will contribute to inhibit and regulate the caspases. Therefore, TNF stimulation triggers two opposite effects: activation and inhibition of caspases. The dynamics of the pro- and anti-apoptotic pathways, as well as the interconnections between them, will ultimately lead to a decision between cell death or cell survival. An abundance of active caspases (such as C3a) together with a low concentration of IAP typically leads to cell death. In contrast, a high concentration of IAP and a low level of active caspases typically characterizes a living cell (in this case, enough molecules IAP are present to down-regulate the level of active caspases). In the network represented in Fig. 1, and in particular in its corresponding Boolean model (Table 1), steady states corresponding to cell death should satisfy C3a = 1 and IAP = 0, while steady states corresponding to cell survival should satisfy C3a = 0 and IAP = 1.

The Boolean model in Table 1 has been slightly simplified from that in [6], namely the mRNAs have been removed and only the corresponding proteins nodes are represented. This does not affect the overall dynamics, but reduces the number of variables to facilitate the use of the asynchronous algorithms. The analysis of the transition graph of the Boolean network will allow us to study the dynamics of the system, in particular the effect of the structure of the network in creating and/or maintaining a balance between the pro- and anti-apoptotic pathways, and ultimately the decision between death or survival.

3. Hierarchical organization of the asynchronous transition graph

In this section, a general methodology to analyze the asynchronous transition graph of a Boolean network is presented. This methodology is based on different algorithms that are classical in the field of graph theory (mainly the strongly connected components decomposition and the topological sort). One can refer to [7] for a detailed analysis of these algorithms. Various implementations of strongly connected components decomposition and hierarchical organization, dedicated to biological networks, already exist in the literature (see, for instance, [11, 4]). We provide an alternative implementation of these algorithms, specifically designed to handle Boolean models of biological genetic regulatory networks (where a node has only two values). Our goal is to make an optimal use of the hierarchical organization of the transition graph in order to answer, algorithmically, some specific biological questions, related to local dynamical behaviors of the system. For instance, given any state, what regions of the phase space lead, or not, to it? What attractors are reachable, or not, from it? Moreover, in Section 4, we will also

Node	Boolean rule
TNF	TNF (input of the whole system)
T2	$\mathrm{TNF} \wedge \overline{\mathrm{FLIP}}$
IKKa	$\mathrm{TNF} \wedge \overline{\mathrm{A20a}} \wedge \overline{\mathrm{C3a}}$
NFκB	$\overline{\mathrm{I}\kappa\mathrm{B}}$
NFκB$_{\mathrm{nuc}}$	$\mathrm{NF}\kappa\mathrm{B} \wedge \overline{\mathrm{I}\kappa\mathrm{B}}$
IκB	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{IKKa}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{IKKa}})]$
A20a	$\mathrm{TNF} \wedge \mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}}$
IAP	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{C3a}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{C3a}})]$
FLIP	$\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}}$
C3a	$\overline{\mathrm{IAP}} \wedge \mathrm{C8a}$
C8a	$\overline{\mathrm{CARP}} \wedge (\mathrm{C3a} \vee \mathrm{T2})$
CARP	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{C3a}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{C3a}})]$

Table 1: Boolean rules for the apoptosis network depicted in Fig. 1. See explanation of variables in the text. Note that the variable TNF can be considered an input of the system, as its activation function is TNF′ = TNF.

provide an algorithmic method, based on the hierarchical organization, to further analyse the system and identify a subnetwork of interactions responsible for such local behaviors.

3.1. SCC decomposition and hierarchical organization

The notion of hierarchical organization of a directed graph (or *digraph*) relies on the well known strongly connected components (SCC) decomposition algorithm. Let us first recall some basics about digraphs (see [7] for more details). Let $G = (V, E)$ be a digraph. Two vertices $u, v \in V$ are *mutually reachable* (denoted $u \sim v$) if and only if there exist two (directed) paths ρ and ρ' such that ρ joins u to v and ρ' joins v to u. This relation is clearly an equivalence relation on the set V of vertices. The *strongly connected components* of the digraph G are then defined as the elements of $V/\sim$, that is to say the equivalence classes of the relation $\sim$. In other words, a strongly connected component of G is a maximal set of vertices $C \subseteq V$ such that for every pair $u, v \in C$, u and v are reachable from each other.

The SCC *decomposition* of a digraph G consists in computing the strongly connected components of G: $C_1, \ldots, C_p$ and then to compute the digraph $G^{scc} = (V^{scc}, E^{scc})$ defined as follows:

- $V^{scc} = \{C_1, \dots C_p\}$,
- given $1 \leq i, j \leq p$, the directed edge (C_i, C_j) belongs to E^{scc} if and only if there are $u \in C_i$ and $v \in C_j$ such that $(u, v) \in E$.

It can be easily proved (see [7]) that the digraph G^{scc} contains no (oriented) cycles. It is called a *dag* (for directed acyclic graph). This is a key property of G^{scc}, because every dag can be *topologically sorted* (see [7], section 22.4). A topological sort of a dag can be viewed as a classification of its vertices in several hierarchical levels $H_1, H_2, \dots$, defined, by induction, as follows:

- $H_1 = \{C_q \in V^{scc} : \forall C_{q'} \in V^{scc}, (C_{q'}, C_q) \notin E^{scc}\}$,
- $H_i = \{C_q \in V^{scc} : \forall C_{q'} \in V^{scc}, ((C_{q'}, C_q) \in E^{scc} \Rightarrow \exists j < i, C_{q'} \in H_j)\}$,

so that the vertices of the first level H_1 are vertices with no predecessors, and the predecessors of vertices of level H_i, $i > 0$, are contained in inferior levels H_j with $j < i$ (see also Fig. 2). The decomposition and hierarchical organization of a digraph G can be computed in linear time with respect to the number of vertices and edges of G [7]. The main interest of this hierarchical organization, applied to the asynchronous transition graph of a Boolean network, is that, whatever path we choose in the graph (*i.e.* whatever updating order we choose for the variables), once the path leaves a hierarchical level H_i, it cannot return to this level. So, any path will travel "down" the hierarchical levels: $H_{i_1} \to H_{i_2} \to \dots$ (with $i_1 < i_2 < \dots$). Due to this property, we can now give a precise definition of the term *attractor* for a Boolean network evolving according to an asynchronous strategy.

Definition 4 (Attractor) *Let $\mathcal{N}$ be a Boolean network. An SCC $c^* \in V^{scc}$ that has no successor in G^{scc} is called an (asynchronous) attractor of $\mathcal{N}$.*

In graph theory, such SCCs are often called *terminal* SCCs. In other words, the asynchronous attractors of a Boolean network are the strongly connected components of the transition graph that cannot be escaped by the system, whatever the updating strategy. However, it should be noted that it is still possible to construct specific asynchronous updating strategies such that the system gets "stuck" in a non terminal SCC. Indeed, for any SCC that contains at least two states, it is obvious that we can find a particular strategy that allows the system to remain indefinitely in this component (by strong connectedness). Such intermediate SCCs will not be considered as attractors in this paper, as we seek general dynamical properties that are valid for all the choices of updating rules. The hierarchical

organization of the transition graph allows the formulation of simple algorithmic definitions of attraction and reachability sets.

Algorithm 1 - Computation of attraction and reachability sets in G^{scc}.

`Input :` $c \in \{1, \ldots, p\}$ (a SCC of the asynchronous transition graph).
`Output :` $\mathcal{A}(c), \mathcal{R}(c)$: attraction and reachability sets of c.

1: $\mathcal{A}(c)$:= ATTR(c)
2: $\mathcal{R}(c)$:= REACH(c)

where ATTR and REACH are two simple recursive functions:

1: **ATTR**(c):
2: $A := \{c\}$
3: P := predecessors(c)
4: **if** $P \neq \varnothing$ **then**
5: **for all** $\gamma \in P$ **do**
6: $A := A \cup$ ATTR(γ)
7: **end for**
8: **end if**
9: **return** A

1: **REACH**(c):
2: $R := \{c\}$
3: S := successors(c)
4: **if** $S \neq \varnothing$ **then**
5: **for all** $\gamma \in S$ **do**
6: $R := R \cup$ REACH(γ)
7: **end for**
8: **end if**
9: **return** R

The functions *predecessors* and *successors* return, respectively, the -possibly empty- sets of immediate predecessors and successors of a node in the dag G^{scc}. As explained in the text, if the node $\gamma \in V^{scc}$ belongs to a hierarchical level H_i, then its predecessors (resp. its successors) can only lie in hierarchical levels H_j with $j < i$ (resp. with $j > i$).

Definition 5 *Let $c \in V^{scc}$ be a SCC of the asynchronous transition graph. The sets $\mathcal{A}(c)$ and $\mathcal{R}(c)$ computed by Algorithm 1 are, respectively, the attraction set of c (i.e. the set of all SCCs that can lead to c) and the reachability set of c (i.e. the set of all SCCs that can be reached from c). If c is an attractor of the network (in other words, if $\mathcal{R}(c) = \{c\}$), the set $\mathcal{A}(c)$ is its basin of attraction.*

The definition of attraction or reachability set should be understood as a *weak* notion, in the sense that the existence of a trajectory from a SCC c to a SCC c' is sufficient to have $c \in \mathcal{A}(c')$ (or $c' \in \mathcal{R}(c)$). On the contrary, Definition 4 of an attractor c^* is *strong*, in the sense that no trajectories are allowed to move out of c^*. These notions of *strong* and *weak* attraction are remniscent of those found in [3], in the context of equilibria of differential inclusions. In [3] a set

of equilibria is said to be weakly asymptotically stable if there is *at least one solution* of the differential inclusion for which the set is asymptotically stable in the classical sense. The strong notion holds if the set is asymptotically stable in the classical sense *for every solution* of the differential inclusion. In a discrete graph, if $a_1, \ldots, a_r$ designate the attractors of the network, an element $c \in \mathcal{A}(a_1)$ may lead to attractor a_1 (*i.e.*, there exists an updating strategy such that c leads to a_1). If one wants the basin of attraction of a_1 in a strong sense, that is, the set of SCCs that *always* lead to a_1 (whatever the updating order), then one has to compute the set:

$$\mathcal{A}^s(a_1) = \mathcal{A}(a_1) \setminus \left(\bigcup_{i=2}^{r} \mathcal{A}(a_i) \right),$$

which may in some cases be reduced to the singleton $\{a_1\}$.

3.2. Application to the apoptosis network

The SCC decomposition and hierarchical organization were applied to the NFκB signalling pathway described in Section 2.3. The results presented here were obtained with codes implemented in `Matlab`. Following the `matlab_bgl`[3] library specifications, the graphs are represented with sparse matrices, which allow a quite efficient implementation.

We recall that the system under study is of dimension $n = 12$, and that one particular variable, TNF, is an input (*i.e.* its activation function is TNF′ = TNF). The state space is $\Omega = \{0, 1\}^n$, and the size of the asynchronous transition graph G is $2^n = 4096$. The number of strongly connected components is $p = 1472$, therefore the size of the graph G^{scc} is only 40% of the size of G. After the hierarchical organization of this graph, we found only 38 hierarchical levels, and 3 attractors. Fig. 2 represents a scheme of this graph with its main elements. The fact that TNF is an input implies that its value remains constant, whatever path is chosen. Mathematically, this means that G^{scc} is the union of two disconnected (sub)graphs, denoted $\mathcal{T}^0$ (where TNF = 0) and $\mathcal{T}^1$ (where TNF = 1). The components $\mathcal{T}^0$ and $\mathcal{T}^1$ are two subsets of nodes of G^{scc} that are completely separated (there exist no directed edge going from a SCC in $\mathcal{T}^0$ to a SCC in $\mathcal{T}^1$, and vice versa).

We found three terminal SCCs in our graph, which means that the system has three different attractors. Using the SCC labels returned by the hierarchization, the SCCs 1 and 123 contain only one state (they are therefore steady states) whereas

[3] see `http://www.stanford.edu/~dgleich/programs/matlab_bgl/`

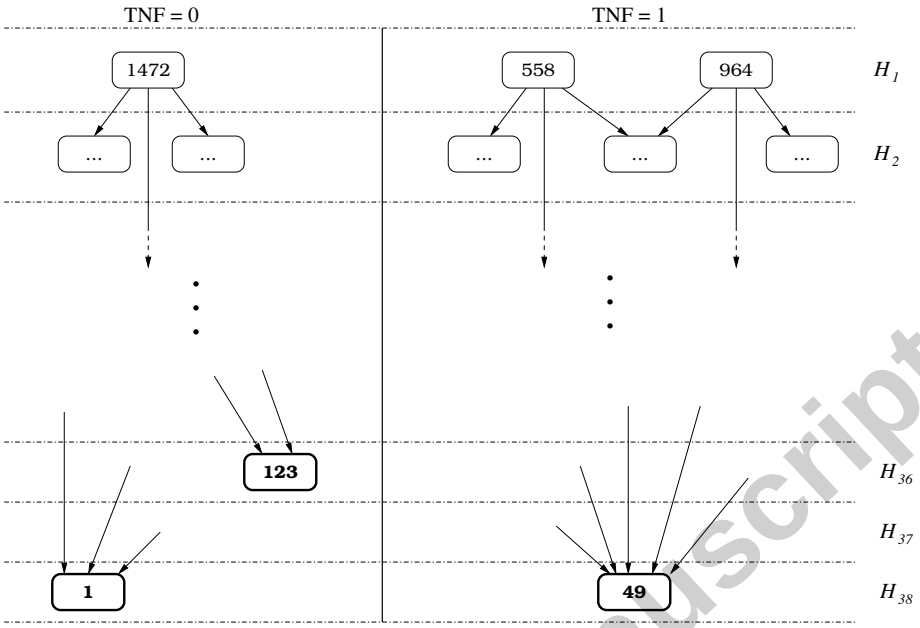


Figure 2: Scheme of the main elements of the hierarchical graph G^{scc} for the apoptosis network. The vertical line separates the two connected components (generated by the input TNF), and the horizontal lines separates the different hierarchical levels. The SCCs are designated by their integer index (between 1 and $p = 1472$). The only SCCs that are represented here are the roots (SCCs that belong to the first hierarchical level) and the attractors (in bold characters).

the third one, 49 is a more complex SCC with 56 states. Table 2 indicates the Boolean values taken by the variables within each attractor. The two steady states belong to $\mathcal{T}^0$ (TNF is absent), the first one corresponds to survival of the cell (the caspases C3a and C8a are absent) and the second one corresponds to the triggering of apoptosis (with activation of the caspases). The complex attractor (SCC 49) belongs to $\mathcal{T}^1$ (TNF is present). As we can see in Table 2, within this attractor the caspases are activated while NFκB, IκB and other factors oscillate. At first, this might seem to indicate that apoptosis will be the final outcome, but, as will be seen later, upon TNF removal, the cell may still choose either the survival or apoptotic steady states.

In the previous example, the computations of the asynchronous transition graph and of its hierarchical organization take only a few seconds (simulations made on a PC: Intel 2.13 GHz, RAM: 1.99 Go). The implementation of these graphs is based on sparse matrices, which makes the storage of data and the run

	TNF	T2	IKKa	NFκB	NFκB$_{nuc}$	IκB
Attractor a_L (SCC 1)	0	0	0	0	0	1
Attractor a_D (SCC 123)	0	0	0	0	0	1
Attractor a_O (SCC 49)	1	*	0	*	*	*
	A20a	IAP	FLIP	C3a	C8a	CARP
Attractor a_L (SCC 1)	0	1	0	0	0	1
Attractor a_D (SCC 123)	0	0	0	1	1	0
Attractor a_O (SCC 49)	*	0	*	1	1	0

Table 2: Boolean patterns of the three attractors. Attractor a_L is a steady state that corresponds to cell survival (or "life"). Attractor a_D is another steady state that corresponds to the triggering of apoptosis (or "death"). Attractor a_O is a SCC which contains 56 states. The symbol * means that the corresponding variable has no fixed value in the attractor and may oscillate between 0 and 1.

of the different algorithms rather efficient (including the computation of attraction and reachability sets). However, as the size of the transition graph is 2^n, the time complexity of its construction grows exponentially. Moreover, the non determinism of the graph makes its analysis more difficult than in the synchronous case (in particular in the search of attractors, or attraction basins). At present, with not fully optimized codes, the computation time remains reasonable for dimensions n around 15. This is a major difference with the synchronous framework, where the determinism of the transition graph makes it possible to analyze much higher dimensional systems [37].

The hierarchical organization has the advantage of characterizing the dynamics *for all* possible updating orders. It allows for instance to detect "spurious" behaviors, that may appear when the network get apparently stuck in a non terminal SCC. It also comprises all the possible dynamical behaviors in a finite structure, avoiding generation of large numbers of simulations.

4. Identification of operational interactions

The family of SCCs and their transition graph describe a new state space (reduced from 2^n to p states), which characterizes the dynamics of a new, reduced, system. A second stage in our model reduction procedure involves the determination of a set of rules governing this new system. Starting from the simplified asynchronous transition graph G^{scc}, the goal of this part is to reconstruct, as far as possible, the *active*, or *operational* interactions along time (the definition of an

operational interaction can be found in the following). Recall that the SCCs are hierarchically organized, in such way that trajectories can go in one sense from one level to another. Thus, the G^{scc} graph is particularly well suited for an identification process which consists, roughly, of finding all the operational interactions from a level l down to the terminal level (*i.e.* down to the possible attractors).

In general, this identification method can be applied to any, hierarchically organized, state space. It can be used to uncover groups of variables (and interactions) that are mainly responsible for the dynamical behavior of the system in a given region of the state space. More precisely, the method identifies groups of interactions responsible for the system's asymptotic behaviour, from a given level in the state space, or within a "self-contained" subgraph.

4.1. Synchronous identification algorithm

The identification technique that will be used consists in the Boolean identification algorithm REVEAL developed in [20]. The term *identification* must be understood in a precise sense, related to the Boolean structure of the networks under study. Basically, given a family of transition pairs $\{s_i \rightarrow s_j\,,\ s_i, s_j \in \Omega\}$, one wants to *reconstruct* the structure of a network, that is: its interaction graph $\mathcal{G}$ and (possibly) its set of Boolean rules $\mathcal{F} = \{f_i,\ i = 1 \dots n\}$. This algorithm is based on the fact that data (typically, time series issued from DNA microarrays), are supposed to be *synchronous*. More precisely, data consist in temporal sequences of Boolean vectors, $(X^t)_{t=0,1,\dots}$, that satisfy, for all t and i, $X_i^{t+1} = f_i(X^t)$, where the f_i are the functions that we seek to identify. Details about REVEAL can be found in [20, 36] and references therein. In order to describe our method, we summarize in the following some basic facts.

Definition 6 *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ and $j \in \{1,\dots,n\}$. The variable x_j will be called fictitious in function f if, for all $(x_1,\dots,x_{j-1},x_{j+1},\dots,x_n) \in \{0,1\}^{n-1}$,*

$$f(x_1,\dots,x_{j-1},0,x_{j+1},\dots,x_n) = f(x_1,\dots,x_{j-1},1,x_{j+1},\dots,x_n)$$

The set of indices j such that x_j is non-fictitious in f is called the support of f and is denoted $\sigma(f)$.

The *real* connectivity of f, as evoked in Rem. 1, is in fact the size of $\sigma(f)$. We will also use the following notation. For $1 \leq k \leq n$, let $I = \{i_1,\dots,i_k\} \subseteq \{1,\dots,n\}$ be a set of indices and let $\bar{I} = \{j_1,\dots,j_{n-k}\}$ denote the set $\{1,\dots,n\}\backslash I$. For each $x = (x_1,\dots,x_n) \in \{0,1\}^n$, x_I (respectively $x_{\bar{I}}$) designates the k-tuple $(x_{i_1},\dots,x_{i_k})$

(respectively the $(n-k)$-tuple $(x_{j_1},\ldots,x_{j_{n-k}})$). If x_I is fixed in $\{0,1\}^k$, f_{x_I} will designate the function:

$$\begin{array}{rccc} f_{x_I}: & \{0,1\}^{n-k} & \longrightarrow & \{0,1\} \\ & (x_{j_1},\ldots,x_{j_{n-k}}) & \longmapsto & f(x_1,\ldots,x_n). \end{array}$$

This lemma will be useful in the following:

Lemma 1 *The support of f satisfies $\sigma(f) \subseteq I$ if and only if, for all $x_I \in \{0,1\}^k$, the function f_{x_I} is constant.*

The proof, that comes directly from Def. 6, is left to the reader.

The data used by REVEAL, given by a series of (input state,output) pairs, are first arranged under the form of a truth table, defined as a couple (In, Out) of $q \times n$ Boolean matrices such that the rows of In are all distinct ($1 \leq q \leq 2^n$). The purpose of REVEAL is to find sets of Boolean functions that are *consistent* with (In, Out), as defined by:

Definition 7 *Let In and Out be two $q \times n$ Boolean matrices, such that all the rows of In are different. A Boolean function $f_j : \{0,1\}^n \to \{0,1\}$ is consistent with (In, Out) with respect to column j if, for all row $R_i(In)$ of In ($i = 1 \ldots q$), we have $f_j(R_i(In)) = Out_{ij}$. A function $F = (f_1,\ldots,f_n) : \{0,1\}^n \to \{0,1\}^n$ is consistent with (In, Out) if for all j, f_j is consistent with respect to column j.*

Basically, the main loop of REVEAL consists in finding, for each node v_j of the network, k-tuples of nodes $(v_{i_1},\ldots,v_{i_k})$ that are possible sets of inputs for the logical rule of v_j. Mathematically, the existence of such a tuple implies the existence of a function f_j that is consistent with respect to column j and which satisfies $\sigma(f_j) = \{i_1,\ldots,i_k\}$. To test whether the tuple $(v_{i_1},\ldots,v_{i_k})$ is a possible set of inputs for v_j, the following equality is checked:

$$H\left(C_{Out}(j), C_{In}(i_1),\ldots,C_{In}(i_k)\right) = H\left(C_{In}(i_1),\ldots,C_{In}(i_k)\right),$$

where $C_X(i)$ designates the i-th column of matrix X, and H designates the Shannon's entropy (a classical notion in information theory), defined by:

Definition 8 *Let $M_1,\ldots,M_r \in \{0,1\}^q$, and let M designate the $q \times r$ Boolean matrix of columns $M_1,\ldots,M_r$. The entropy of M is the quantity $H(M)$ defined by:*

$$H(M) = H(M_1,\ldots,M_r) = -\sum_{x \in \{0,1\}^r} P_M^x \log(P_M^x),$$

where log *designates the logarithm to the base* 2*, and* $P_M^x \in [0,1]$ *is the proportion of rows of* M *that are equal to the boolean vector* x*.*

To justify the correctness of the algorithm, we prove the following

Proposition 1 *Let* $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ *and* $j \in \{1, \ldots, n\}$*. The two assertions are equivalent:*

(i) $H(C_{Out}(j), C_{In}(i_1), \ldots, C_{In}(i_k)) = H(C_{In}(i_1), \ldots, C_{In}(i_k))$*,*

(ii) There exists a function f_j *that is consistent with* (In, Out) *with respect to column* j *and such that* $\sigma(f_j) \subseteq \{i_1, \ldots, i_k\}$*.*

Proof.
Let M (resp. M') designate the matrix formed by the columns $C_{In}(i_1), \ldots, C_{In}(i_k)$ (resp. by the columns $C_{Out}(j), C_{In}(i_1), \ldots, C_{In}(i_k)$). Definition 8 yields:

$$\begin{cases} H(M) &= -\sum_{x\in\{0,1\}^k} P_M^x \log(P_M^x) \\ H(M') &= -\sum_{x\in\{0,1\}^k} P_{M'}^{0,x} \log(P_{M'}^{0,x}) - \sum_{x\in\{0,1\}^k} P_{M'}^{1,x} \log(P_{M'}^{1,x}) \end{cases}$$

Moreover, we have: $P_M^x = P_{M'}^{0,x} + P_{M'}^{1,x}$. Equality (i) is thus equivalent to:

$$\sum_{x\in\{0,1\}^k} \left[P_{M'}^{0,x}(\log(P_{M'}^{0,x} + P_{M'}^{1,x}) - \log(P_{M'}^{0,x})) + P_{M'}^{1,x}(\log(P_{M'}^{0,x} + P_{M'}^{1,x}) - \log(P_{M'}^{1,x})) \right] = 0$$

As this is a sum of nonnegative terms, it is equivalent to:

$$\begin{aligned} P_{M'}^{0,x}(\log(P_{M'}^{0,x} + P_{M'}^{1,x}) - \log(P_{M'}^{0,x})) &= 0 \\ P_{M'}^{1,x}(\log(P_{M'}^{0,x} + P_{M'}^{1,x}) - \log(P_{M'}^{1,x})) &= 0 \end{aligned}$$

which both imply that either $P_{M'}^{0,x}$ or $P_{M'}^{1,x}$ is equal to zero in each term of the sum, *i.e.* for each $x \in \{0,1\}^k$. In other words, this means that (i) is equivalent to the following assertion:

(iii) For each $x \in \{0,1\}^k$, let $A_x := \{i : \ 1 \leq i \leq q,\ R_i(In) = x\}$ (where $R_i(In)$ denotes the i-th row of In). Then there exists $\alpha_x \in \{0,1\}$ such that $Out_{ij} = \alpha_x$ for all $i \in A_x$.

Suppose (i) (or equivalently (iii)) is satisfied, let us construct a function f_j that satisfies (ii). To do that, take any Boolean vector $x \in \{0,1\}^n$. We have to consider two different cases. First, suppose there exists a row $R_i(In)$ of In such that $In_{i,i_l} =$

x_{i_l}, for all $l = 1, \ldots, k$. Then, set $f_j(x) = Out_{ij}$ (which, according to (iii), is independent of the choice of i). If, on the other hand, the pattern x_I is not present in In, then we have a degree of freedom: we choose $\epsilon(x_I) \in \{0, 1\}$ and set $f_j(x) = \epsilon(x_I)$. By construction, such a function f_j is consistent with (In, Out) with respect to column j. Furthermore, each projection $(f_j)_{x_l}$ is constant which, according to Lemma 1, implies that $\sigma(f_j) \subseteq I$. This proves that (i)⇒(ii). The converse implication can be proved using analogous arguments. □

The main loop of the algorithm is made for increasing k (from $k = 0$, detecting constant functions, to $k = n$). In order to avoid redundancies, the algorithm stops once it finds the minimal connectivity of each node, *i.e.* the minimal value of k that ensures the existence of a consistent function. Therefore, the functions identified by REVEAL are *minimally consistent*, in the sense that, among all consistent functions, they are the ones with a support of minimal size.

As a general reverse engineering algorithm, the purpose of REVEAL is to identify consistent functions "from scratch", that is, independently of any prior knowledge about the system under study. The problem that is addressed here is slightly different, as we are searching subsets of interactions included in an already known set of interactions. Therefore, given a collection of Boolean functions $\mathcal{F} = \{f_1, \ldots, f_n\}$, we force the algorithm to select only the minimally consistent functions g_j whose support is included in $\sigma(f_j)$ (see Alg. 2).

Algorithm 2 - Identification of minimally consistent sub-networks.

`Input:` $(\mathcal{G}, \mathcal{F})$: Boolean network. In, Out: Boolean $q \times n$ matrices.
`Output:` set of sub-networks minimally consistent with (In, Out).

1: mark all nodes $\{v_1, \ldots, v_n\}$ as *untreated*
2: **for** $k = 0, 1, \ldots, n$ **do**
3: **for all** *untreated* node v_i **do**
4: **for all** $(i_1, \ldots, i_k) \subseteq \sigma(f_i)$ **do**
5: **if** $H(C_{Out}(i), C_{In}(i_1), \ldots, C_{In}(i_k)) = H(C_{In}(i_1), \ldots, C_{In}(i_k))$ **then**
6: $\rightarrow (v_{i_1}, \ldots, v_{i_k})$ are inputs of node v_i
7: $\rightarrow$ mark node v_i as *treated*
8: **end if**
9: **end for**
10: **end for**
11: **end for**

For the original REVEAL algorithm, the existence of a minimally consistent function is guaranteed, but this is not the case for Algorithm 2. In Section 4.2, we propose a way to construct truth tables (In, Out) that will ensure the existence of solutions. Regarding uniqueness, the proof of Prop. 1 shows that if a solution exists, it is not necessarily unique if the size of the table q is $< 2^n$.

4.2. Algorithmic search for operational interactions

In order to properly define the algorithmic search for operational interactions, let us first recall the following definitions, classical in graph theory.

Definition 9 *Let $G = (V, E)$ be a directed graph.*

- *For $V' \subset V$, the* subgraph *of G, induced by V', is the graph $G(V') = (V', E(V'))$, where $E(V')$ is the subset of E containing only the directed edges whose head and tail both belong to V'.*
- *For $E' \subset E$, the graph $G' = (V, E')$ is called a* partial graph *of G (G' is said to be* included *in G, denoted by $G' \subset G$).*

Let $\mathcal{N} = (\mathcal{G}, \mathcal{F})$ be a (given) n-dimensional Boolean network, and let $G = (\Omega, E)$, $G^{scc} = (V^{scc}, E^{scc})$ denote, respectively, its asynchronous transition graph and its (hierarchically organized) SCC decomposition. We will also make use of the synchronous transition graph of the system, which will be designated by $G^{sync} = (\Omega, E^{sync})$ (recall that each state $s \in \Omega$ has a unique successor in that graph, which is the state $s' = (s'_1, \ldots, s'_n)$ where $s'_i = f_i(s)$). If $c \in V^{scc}$ denotes a SCC of G, Algorithm 1 computes the reachability set of c, $\mathcal{R}(c)$. This set contains all SCCs (and thus all states) that are reachable by the system starting from c, whatever the updating order of the variables. Therefore, it is possible to construct the subgraph of G^{scc} (respectively, the subgraph of G), induced by $\mathcal{R}(c)$, that contains all possible SCC trajectories starting from c (resp., all possible state trajectories starting from any state in c). Let $G^{scc}(c)$ (resp., $G(c)$) denote this subgraph. Using Assumptions 1 and 2, it is straightforward to deduce, from $G(c)$, the subgraph $G^{sync}(c)$ of G^{sync} induced by $\mathcal{R}(c)$. This subgraph $G^{sync}(c)$ is then organized under the form of a truth table (In, Out) (see Algorithm 3 - step 1). The application of Algorithm 2 on these synchronous data (step 2 of Alg. 3) allows identification of all interactions which are sufficient to reproduce all asynchronous trajectories issued from c. These interactions are called *operational* and can be formally defined as follows:

Definition 10 *Consider an n-dimensional Boolean network with interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $G = (\Omega, E)$ be its asynchronous state transition graph. Let $c \subset \Omega$ and let $\mathcal{R}(c)$ denote the set of states reachable from any state in c. The subgraph of G induced by $\mathcal{R}(c)$ is designated by $G(c)$.*

- *An edge $e \in \mathcal{E}$ is a* non-operational interaction associated with *c if the asynchronous subgraph $\hat{G}(c)$, generated by $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \setminus \{e\})$ (starting from c), satisfies: $\hat{G}(c) = G(c)$.*

- *An edge $e \in \mathcal{E}$ is an* operational interaction associated with *c if e is not non-operational.*

Let $\mathcal{E}_c$ denote the set of operational interactions, the interaction graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ will be called operational graph.

In other words, a minimal family of operational interactions (for the transition graph generated by a set of states c) contains as small a set as possible of the original interactions that still generates the original graph $G(c)$. The term *minimal* refers to the fact that, thanks to REVEAL, the supports of the identified functions are of minimal size.

From the construction of the table (In, Out), it is obvious that Alg. 2 effectively returns a solution (in the worst case, it will return the original functions $f_1, \ldots, f_n$, that are obviously consistent with (In, Out)). However, as already evoked, the returned solution is not necessarily unique. Nevertheless, for the apoptosis network treated in this paper, the operational graph is in fact unique for any SCC c (see Section 4.3). As the reachability set $\mathcal{R}(c)$ captures all possible asynchronous successors of c, it is easy to see that, if we successively compute the operational graphs along a particular SCC trajectory: $(c_1, c_2, \ldots, c_l)$ (where c_l is an attractor of the system), then we have the following inclusions:

$$\mathcal{G} \supset \mathcal{G}_{c_1} \supset \mathcal{G}_{c_2} \supset \cdots \supset \mathcal{G}_{c_l}.$$

(Symbol $\subset$ designates the graph inclusion defined in Def. 9). In other words, this means that it is possible to visualize, along a trajectory, at which step an interaction (represented by a directed edge of $\mathcal{G}$) may become non-operational. Ultimately, the final graph $\mathcal{G}_{c_l}$ comprises interactions that remain always operational through the whole trajectory (up to the attractor).

A related notion is that of *minimal cut sets* (MCS) for logical interaction graphs, suggested and used in [19]. A MCS has been defined with respect to a

Algorithm 3 - Identification of asynchronous operational interactions.

Step 1: Construction of synchronous data from a subgraph Γ of the asynchronous transition graph G.

`Input :` $\Gamma = (V(\Gamma), E(\Gamma))$: subgraph of G.
`Output :` In, Out: Boolean matrices of size $q \times n$ (partial truth tables).

1: $q := 1$
2: **for all** state $X \in V(\Gamma)$ **do**
3: $In(q,:) := X$ /* fill in the q-th row of In */
4: $S := X$ /* will contain the synchronous successor of X */
5: **for all** asynchronous successors Y of X (in Γ) **do**
6: find i such that $Y = \widetilde{X}^i$
7: $S(i) := not(X(i))$
8: **end for**
9: $Out(q,:) := S$, $q := q + 1$
10: **end for**

Step 2: Identification of operational interactions from a SCC c.

1: compute reachability set $\mathcal{R}(c)$ /* use Algorithm 1 */
2: compute subgraph $G^{scc}(c)$ of G^{scc} induced by $\mathcal{R}(c)$
3: compute corresponding subgraph $G(c)$ of G
4: compute synchronous truth table (In, Out) from $G(c)$ /* use step 1 */
5: compute operational graphs /* use Alg. 2 */

certain target function or response, and by analogy with the analysis of (continuous, stoichiometric) metabolic networks. A MCS is a minimal set of reactions whose removal will prevent the target response. The notion of operational interaction defined here differs from that of MCS. Mainly, the operational graph identifies a subnetwork responsible for a certain dynamical behaviour, that is comprised in a certain region of the transition graph. The reduction of the whole state space to this particular region allows to ensure all non-operational interactions to be removed. In contrast, MCS are not defined in terms of transition graphs.

4.3. Application to the apoptosis network

In this section, Algorithm 3 is applied to the apoptosis NFκB signalling pathway. This system is interesting because it exhibits two global dynamical properties that are often studied in systems biology. The first one is the *multistationarity*,

i.e. the coexistence of several attractors. According to Fig. 2, this happens when TNF is absent (the two attractors are steady states). The second one is the presence of *oscillations*, which appear when TNF is activated. These two general properties are often related to the interaction graph, and in particular to the presence of positive and negative feedback loops. Two famous conjectures, stated in the eighties by R. Thomas [33], have been proved in different mathematical frameworks (see for instance [14, 32, 24]). The first one states that the presence of positive feedback loops is a necessary condition for multistationarity. The second one states that the presence of negative feedback loops is a necessary condition for the presence of oscillations (in continuous frameworks, oscillations may be damped, and can then be related to the biological concept of *homeostasis*). In the following, Alg. 3 is used to identify which feedback loops of Fig. 1 are effectively responsible for the presence of oscillations and for multistationarity.

According to the analysis presented in Section 3.2, the state space of the apoptosis NFκB system can be separated in two regions, $\mathcal{T}^0$ and $\mathcal{T}^1$, according to the value of the input TNF. Within region $\mathcal{T}^1$ (TNF = 1), we found a unique attractor, a_0 (see Fig. 2). It contains 56 states, and the variables that can oscillate within a_0 are indicated in Table 2. As it is the only attractor present in $\mathcal{T}^1$, we know that all trajectories starting from any state of $\mathcal{T}^1$ will eventually reach it and remain in it for all subsequent times. When applying Algorithm 3 to a_0, we obtain the operational graph $\mathcal{G}_{a_0}$ depicted in Figure 3.

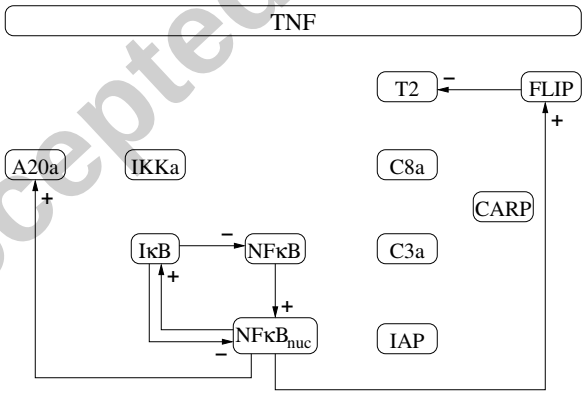


Figure 3: Operational graph of the NFκB pathway in attractor a_0 (TNF = 1). The isolated variables have a fixed value, that can be found in Table 2.

This interaction graph is included in the initial one (Fig. 1), and comprises all interactions that remain active in a_0. As we can see in Fig. 3, six of the twelve

variables are isolated, that is, they keep a constant value over time. Their values can be found in Table 2 (in particular, the caspases C3a and C8a are activated). Among the six remaining variables, only three are involved in feedback loops: IκB, NFκB and NFκB$_{\text{nuc}}$ (the three other variables are affected by those three, but do not affect them). One can see that both feedback loops: NFκB$_{\text{nuc}} \leftrightharpoons$ IκB and IκB $\rightarrow$ NFκB $\rightarrow$ NFκB$_{\text{nuc}} \rightarrow$ IκB are negative, as they contain an odd number of inhibitions. Thomas' conjectures [14, 32, 24] lead to the following conclusions: (i) there cannot be more than one steady state because the graph has no positive loop, and (ii) oscillations are possible because the graph has negative loops. This is indeed what is observed in the system. As these loops are the only ones that are active in attractor a_0, it can be infered that they are the ones responsible for the presence of oscillations in the system.

Let us now consider the region $\mathcal{T}^0$, where TNF = 0. As we can see in Figure 2, this region contains two attractors, corresponding to two steady states. The first steady state (a_L) corresponds to the "survival" of the cell, with the inhibition of the caspases, whereas the second one (a_D) corresponds to the triggering of apoptosis, with activation of the caspases. In order to find the feedback loop that is responsible for the coexistence of these two steady states, we have to find a SCC c^* that can lead to both a_L and a_D, *i.e.*:

$$c^* \in \mathcal{A}(a_L) \cap \mathcal{A}(a_D)$$

(where $\mathcal{A}(a_L)$ and $\mathcal{A}(a_D)$ are the attraction sets of a_L and a_D). There are several possible choices for c^*. As we only want to identify the loop responsible for the coexistence of the two steady states, we choose c^* to be the one with the highest hierarchical level. The operational graph $\mathcal{G}_{c^*}$ obtained is depicted in Figure 4.

Note that this graph contains only one feedback loop: C3a $\leftrightharpoons$ C8a, which is a classical double activation system, involving the caspases. From Thomas' conjectures, there can be no oscillations, only multistationarity. This is indeed the case, as there are exactly two steady states: C3a = C8a = 1 (apoptosis) and C3a = C8a = 0 (survival).

These two examples show that it is algorithmically possible, in the Boolean framework, to identify, in a complex interaction graph such as the one in Fig. 1 (comprising multiple positive and negative loops), subsets of loops which are effectively responsible for those two dynamical behaviors.

Each of the two operational graphs represents two different biological scenarios. The first operational graph (Fig. 3), represents the interactions that remain

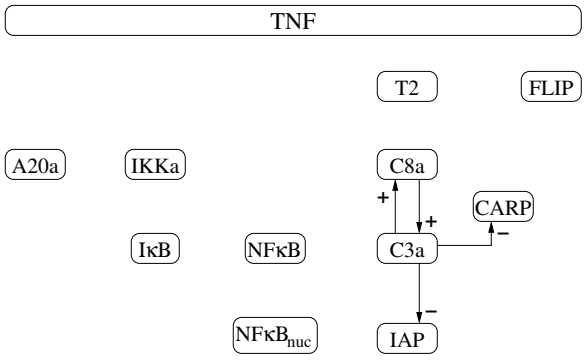


Figure 4: Operational graph of the NFκB pathway just before the choice between attractors a_L (survival) and a_D (apoptosis) (*i.e.* in the region where TNF= 0). Interactions have been signed with respect to Boolean rules (Table 1).

active after a *sufficiently long time interval of TNF stimulation has elapsed*. That is, immediately upon TNF stimulation, the system responds and evolves towards a certain configuration; once this has been reached, most interactions have been "stabilized" or achieved a natural balance. At this stage only remain active the cycle corresponding to NFκB-activated transcription of IκB, and the subsequent inhibition of NFκB. In this case, oscillations in these two variables may be observed [16], but all the other variables are constant. In particular, the model predicts that inhibitor of apoptosis proteins (IAP) is present at a low concentration, but the caspases at high concentration so, from the biological point of view, it may be expected that a very long stimulation with TNF will eventually lead to apoptosis. However, at this stage, the system may still "reverse" its apoptotic decision if TNF is shut down: the system will then leave the configuration shown in Fig. 3, and postpone the survival/death decision. In the absence of TNF, the system will evolve towards the configuration shown in Fig. 4, where only the positive feedback cycle representing the caspase cascase [9] remains functional. From the biological point of view this means that, depending on the state of the system when TNF was shut down (or its initial state), the cell may decide between survival (attractor a_L) or initiating apoptosis (attractor a_D).

5. Probabilistic analysis of asynchronous dynamics

The asynchronous transition graph of a given network is a very general object, in the sense that it contains information on all the possible trajectories of the system: that is, for each state, the graph indicates every possible successor. However,

the graph does not contain any indication on which of the successors is more probable at a given time. Probabilistic approaches to study trajectories of Boolean networks have been used before, under different modelling ideas. For instance, [30] introduces Probabilistic Boolean Networks (PBNs), where several families of logical rules are available ($X' = F^r(X)$, $r \in \{1, \ldots, R\}$). At each updating instant, the family R to be used is chosen according to a probability distribution, but once F^r is chosen the updating scheme is synchronous. In [5, 11, 13], asynchronous updating schemes for Boolean models (with a single family of logical rules) were developed, based on the nature of the biological process (for example, proteins are always updated at a higher frequency than mRNAs). In [11] the variables are divided into priority classes, and each priority class is updated at its own frequency. The idea of asynchronous updates was coupled to PBNs in [10].

Here, we will use the concept of priority classes, to construct a matrix of transition probabilities for the graph G, thus incorporating further quantitative biological knowledge into discrete dynamics' models. Application of our method to the apoptosis/NFκB network leads to estimation, among other quantities, of the probability of cell survival or apoptosis upon stimulation of death receptors. Such quantities can be compared with experimental data (see also numerical results in [6] and references therein).

5.1. Construction of a probabilistic transition graph

In order to confront the general discrete asynchronous dynamics, given by the transition graph G (see Definition 3) with biological experiments, one has to be a little bit more specific on the updating strategies. Previous algorithms and results are essentially *qualitative*, as they are mainly attached to the structure, without considering updating strategies at all. If one wants to represent the different choices of updating strategies, in a more *quantitative* manner, one way is to introduce probabilities in the transition graph G, and to consider a choice of the updating order of the variables (that is, the choice of a particular trajectory in G) as the choice of a trajectory in a stochastic process.

G can be implemented by means of its adjacency matrix, that will be denoted $A(G)$. Recall that each state of the system is a Boolean vector $X \in \Omega = \{0, 1\}^n$. Such a vector X is unequivocally associated with a unique integer $s(X) \in \{1, \ldots, 2^n\}$ (X is the binary decomposition of $s(X)$), so that in the following, "states" (*i.e.* elements of Ω) will be represented by integers lying in $\{1, \ldots, 2^n\}$. The adjacency matrix $A(G)$ is defined as follows:

$$A(G) = \left(a_{i,j}\right)_{1 \leq i,j \leq 2^n}, \text{ with } \begin{cases} a_{ij} = 1 & \text{if "state } j\text{" is a successor of "state } i\text{",} \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$

The size of $A(G)$ is $2^n \times 2^n$ and, in general, $A(G)$ is implemented as a sparse matrix. As a first example, we propose a *naive* construction of the transition probabilities p_{ij}, where all asynchronous successors of a state have the same probability. The uniform distribution of probabilities of asynchronous successors means that we make no *a priori* assumption about the updating rules (beyond of course Assumptions 1 and 2). Define the matrix $\mathcal{P}(G)$ as the $2^n \times 2^n$ real matrix:

$$\mathcal{P}(G) = \left(p_{i,j}\right)_{1 \leq i,j \leq 2^n}, \text{ with: } \quad \forall 1 \leq i, j \leq 2^n, \; p_{ij} = \frac{a_{ij}}{\sum_{k=1}^{2^n} a_{ik}} = \frac{a_{ij}}{N^+(i)},$$

where $N^+(i) = \sum_{k=1}^{2^n} a_{ik}$ designates the number of directed edges of G that leave i (in other words, $N^+(i)$ is the number of asynchronous successors of state i), and $p_{ij} \in [0, 1]$ is the probability for the system to go from state i to state j. By construction, matrices $\mathcal{P}(G)$ and $A(G)$ share the same sparsity pattern. Furthermore, it is easy to see that $\mathcal{P}(G)$ is a *stochastic* matrix, *i.e.* all its elements p_{ij} are nonnegative, and the sum of the elements of each row, $\sum_{j=1}^{2^n} p_{ij}$ is equal to 1.

From the construction of $\mathcal{P}(G)$, we can now consider the (asynchronous) dynamics on G as a discrete time Markov chain, over the state space $\{1, \ldots, 2^n\}$. In particular, if $P(X_0 = i_0)$ designates the probability for the system to be initially in the state $i_0 \in \{1, \ldots, 2^n\}$, then the probability that a trajectory follows the path $\mathbf{p} = (i_0, i_1, \ldots, i_q)$ (where the i_j are elements of $\{1, \ldots, 2^n\}$) is equal to:

$$P(\mathbf{p}) = P(X_0 = i_0) \prod_{j=0}^{q-1} p_{i_j, i_{j+1}}.$$

A similar matrix of probabilities can be constructed for the transition graph G^{scc}, which describes transitions among strongly connected components (SCCs). As explained in Section 3.1, a SCC $c \in \{1, \ldots, p\}$ is, by definition, a subset of the state space Ω. In the following, $S(c)$ denotes the subset of $\{1, \ldots, 2^n\}$ of the states belonging to c, and $|S(c)|$ denotes the size of $S(c)$. The matrix $\mathcal{P}^{scc}$ is a square real matrix of size $p \times p$, and its entries are given by:

$$p^{scc}_{c,c'} = \frac{1}{|S(c)|} \sum_{i \in S(c)} \sum_{j \in S(c')} p_{ij}. \tag{1}$$

In words, this definition means that to compute the probability of the transition from a SCC c to a SCC c', we sum all transitions from any state i in $S(c)$ to any state j in $S(c')$, and we divide this sum by the number of states in c, in order to obtain an average transition probability from c to c'. Moreover, we prove the following:

Proposition 2 *The matrix $\mathcal{P}^{scc}$ is stochastic.*

Proof.
The nonnegativity of $\mathcal{P}^{scc}$ elements is obvious. We prove here that the sum of the elements of its rows is equal to 1. Let $c \in \{1, \ldots, p\}$. We have:

$$\begin{aligned}
\sum_{c'=1}^{p} p^{scc}_{c,c'} &= \sum_{c'=1}^{p} \frac{1}{|S(c)|} \sum_{i \in S(c)} \sum_{j \in S(c')} p_{ij} \\
&= \frac{1}{|S(c)|} \sum_{c'=1}^{p} \sum_{i \in S(c)} \sum_{j \in S(c')} \frac{a_{ij}}{N^+(i)} \\
&= \frac{1}{|S(c)|} \sum_{i \in S(c)} \frac{1}{N^+(i)} \sum_{c'=1}^{p} \sum_{j \in S(c')} a_{ij}.
\end{aligned}$$

As the SCCs of a directed graph form a partition of its set of vertices, the quantity $\sum_{c'=1}^{p} \sum_{j \in S(c')} a_{ij}$ is equal to the number of edges that leave i (in graph G). By definition, it is equal to $N^+(i)$. This leads to:

$$\sum_{c'=1}^{p} p^{scc}_{c,c'} = \frac{1}{|S(c)|} \sum_{i \in S(c)} \frac{N^+(i)}{N^+(i)} = \frac{|S(c)|}{|S(c)|} = 1.$$

□

The main advantage of considering matrix $\mathcal{P}^{scc}$ instead of matrix $\mathcal{P}(G)$ is that it satisfies some useful properties. For instance, each *ergodic* class of $\mathcal{P}^{scc}$ (roughly, an ergodic class is a set of non transient SCCs, see [2] for a precise definition) contains only one element. The corresponding Markov chain is then called *absorbing* and its absorbing elements are in fact the attractors of the system. For absorbing chains, we can use the following well known result (see, *e.g.* [15]): there exists a $p \times p$ permutation matrix P_2 such that

$$P_2 \mathcal{P}^{scc} P_2^t = \begin{pmatrix} I_r & 0 \\ R & Q \end{pmatrix}, \tag{2}$$

where $r \in \{1, \ldots, p\}$ is the number of attractors of the system, I_r denotes the $r \times r$ identity matrix, and Q is a $(p-r) \times (p-r)$ lower triangular matrix that satisfies:

$$\lim_{n \to \infty} Q^n = 0. \tag{3}$$

The form (2) is often called the *canonical form* of $\mathcal{P}^{scc}$. From (3), we can define the $(p-r)\times(p-r)$ matrix $N=(I-Q)^{-1}$ (often called *fundamental* matrix). Its entry $n_{cc'}$ gives the expected number of times that the process is in the transient SCC c' if it started somewhere in the transient SCC c. In particular, it can be used to compute the vector $\mathbf{t}=N\mathbf{1}$ (where $\mathbf{1}$ designates the column vector of whose entries are 1). Given a transient SCC c, the entry t_c of $\mathbf{t}$ gives the expected number of steps before the chain reaches an attractor, given that it started somewhere in c. As a consequence, Markov chains provide an efficient mathematical framework, in which useful global parameters can be computed. These parameters provide important biological knowledge about the system, as they contribute to further characterize qualitative dynamical properties, that are robust with respect to the topology of the network. (See next Sections 5.2 and 5.3.)

5.2. Towards a biological probabilistic graph

The matrix of transition probabilities, $\mathcal{P}(G)$, associated with the asynchronous graph in Section 5.1 was based on a uniform probability distribution. The probabilities of transition can instead be computed according to biological data, by using the notion of *priority classes* [11, 13]. For many biological networks, partial knowledge of the parameters is often available. For example, the relative rates of two reactions are known (*e.g.*, the rate of formation of protein A is larger than that of protein B). This biological knowledge can be straigthforwardly incorporated into the transition graph, by stipulating an updating strategy such as an updating order among all variables. Roughly, the idea is to group the variables into several groups, called priority classes, and assign a weight to each of these groups: higher weights denote a more probable transition. A similar idea was used in [5], where two classes were considered, one for proteins and another for mRNAs. The updating order stipulated that proteins were always updated first and mRNAs next. More generally, to implement the notion of priority classes, consider ρ classes C_1, $\ldots$, $C_\rho\subset\mathcal{V}$ and their respective weights,

$$W_1 > W_2 > \cdots > W_\rho.$$

Recall that, by assumption 1, only one variable is updated at a given instant, so each transition in G involves a 0/1 switch in exactly one variable. Consider an edge $i\to j$ in G, and suppose that $v_s\in\mathcal{V}$ is the node that switches. Then associate with edge $i\to j$ the value:

$$w_{ij}=W_r,\quad \text{if}\quad v_s\in C_r$$

that is, if the node v_s updated in the transition from state i to state j belongs to class C_r. If no transition from i to j is possible, then set $w_{ij} = 0$. Then define a new transition matrix, where each p_{ij} represents a weighted average:

$$\mathcal{P}_{bio}(G) = \left(p_{i,j}\right)_{1\leq i,j\leq 2^n}\ ,\ \text{with:}\quad \forall 1 \leq i, j \leq 2^n\ ,\ p_{ij} = \frac{w_{ij}a_{ij}}{\sum_{k=1}^{2^n} w_{ik}}. \tag{4}$$

As before, a corresponding matrix, $\mathcal{P}^{scc}_{bio}$, can be constructed for the graph G^{scc}. The probability of transition between two SCCs c and c' is given by Eq. (1), where the p_{ij} are replaced by the transition probabilities computed in (4). Again, it is easy to check that $\mathcal{P}^{scc}_{bio}$ represents an absorbing Markov chain process.

5.3. Application to the apoptosis network

In order to illustrate the probabilistic approach, and the type of results it provides, an application to the apoptosis network is next described. In particular, the results show how Tumor Necrosis Factor (TNF) influences the choice of the system between the two possible steady states: the "survival" of the cell (attractor a_L, with inhibition of the caspases) and the triggering of apoptosis (attractor a_D, with activation of the caspases). Experimentally, it is observed that a cell irreversibly enters the apoptotic pathway once a certain threshold in caspase activation has been reached (see [23]). In turn, caspase activation is observed to depend on the duration of TNF stimulation, as well as on TNF concentration (typically, the caspase activation threshold is reached faster for higher TNF concentrations). In [6], this property was statistically observed by computing many different trajectories (of a continuous, piecewise linear system), with different updating strategies. Within the framework presented in this paper, the probability that the cell follows the "survival" or "apoptosis" pathway can be computed directly from the matrices $\mathcal{P}^{scc}$ (or $\mathcal{P}^{scc}_{bio}$), without performing large numbers of simulations. Recall that a trajectory of the system will converge towards a_L or a_D in the absence of TNF. Technically, the shutdown of death receptor stimulation is represented in our system by switching the input variable TNF from 1 to 0. For each state X in $\mathcal{T}^1$ (where TNF is equal to 1), a successor X_s is computed in $\mathcal{T}^0$ (where the variable TNF is 0, and all other variables stay unchanged). Then, using the matrix $\mathcal{P}^{scc}$ and its canonical form (2), the probabilities to reach attractors a_L and a_D from initial state X_s are computed (Figure 5).

Contrary to [6], where time is represented by a continuous variable, in the asynchronous graphs there is no direct measure of time. However, the hierarchical levels of the graph G^{scc} do give an indication of time progression. Indeed,

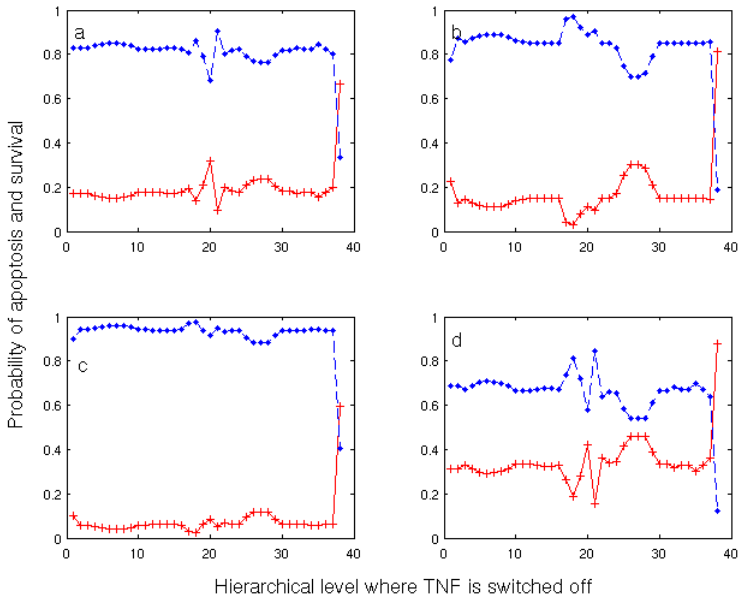


Figure 5: The curves represent the system's response to TNF switch off, after the system has reached a certain hierarchical level H_i (x-axis). The dashed lines with points represent the average probability for the system to reach attractor a_L (the "survival" steady state), starting from a state in hierarchical level H_i. The solid lines with '+' signs represent the probability to reach attractor a_D (the "apoptosis" steady state), starting from H_i.The weights used in each case were: (a) $w_1 = w_2 = w_3 = 1$; (b) $w_3 = 1$, $w_1 = w_2 = 5$; (c) $w_1 = w_3 = 1$, $w_2 = 5$; and (d) $w_2 = w_3 = 1$, $w_1 = 5$.

consider two states along any given trajectory, $X_1, X_2 \in \mathcal{T}^1$ where X_1 belongs to a hierarchical level H_i and X_2 belongs to H_j, with $j > i$. Then we can say that TNF stimulation has been longer for X_2 than for X_1. The x-axis of Figure 5, which represents the hierarchical levels, is thus a relative measure of the duration of TNF.

Following the procedure developed in Section 5.2, a more realistic matrix $\mathcal{P}^{scc}_{bio}$ can be constructed. Based on the parameters reported in [9, 28] (and references therein), and on the results from Section 4.3, three priority classes were established, based on biological function groups (Table 3). The first class represents the negative feedback loops related to the oscillatory behaviour: NFκB, NFκB$_{nuc}$, IκB, and IKKa. The second class represents the caspases' positive loop and its activation: C3a, C8a, and T_2. Finally, the third class contains the remaining proteins (A20a, FLIP, IAP, and CARP) whose transcription is regulated by NFκB$_{nuc}$.

Setting up these classes allowed us to compare the contribution of each family

Class	Weights	Variables
C_1	w_1	NFκB, NFκB$_{nuc}$, IκB, IKKa
C_2	w_2	C3a, C8a, T_2
C_3	w_3	A20a, FLIP, IAP, CARP

Table 3: Priority classes.

of operational interactions to the final survival/apoptosis decision. In a first study, we computed the probabilities of apoptosis and survival, for all possible combinations of $w_i \in \{1, 5\}$ ($i = 1, 2, 3$). This captures the effect of the relative updating frequencies (results are summarized in Table 4). Some examples are shown in Figure 5, which depicts the average probabilities of reaching attractors a_L and a_D, after 50 updating steps with TNF=0,

$$P_L(H_i) = \frac{1}{N_i^1} \sum_{k=1}^{N_i^1} \mathcal{P}_{bio}^{50}(X_k^0, a_L), \quad P_D(H_i) = \frac{1}{N_i^1} \sum_{k=1}^{N_i^1} \mathcal{P}_{bio}^{50}(X_k^0, a_D),$$

where N_i^1 denotes the number of states in $\mathcal{T}^1 \cap H_i$, and $X_k^0 \in \mathcal{T}^0$ denotes the successor of $X_k \in \mathcal{T}^1$ after TNF is switched off (computed as explained above). After 50 steps, the system is likely to have reached an attractor with (practically) 100% probability. Thus, $P_L(H_i) \approx 1 - P_D(H_i)$. In all cases studied, we observe that the probability of survival is larger than that of apoptosis, for *almost all levels* H_i, $i = 1, \ldots, 37$. This situation is reversed for the last level, H_{38}, where the probability of apoptosis is *always* larger than that of survival (see Table 4, column $P_L(H_{38})$). It is striking that this property appears in all cases, suggesting the existence of a threshold state configuration which is independent of the updating strategy, and constitutes a robust feature of the system. In other words, if one wants to promote apoptosis, TNF should be sustained long enough for the system to reach attractor a_0, *i.e.* apoptotic oscillations.

More generally, we observe that the probability of apoptosis (or survival) can be modulated within a certain range of values. At the last level, the probability of apoptosis is above 50%. In the levels H_1 to H_{37}, the probability of reaching a_L dominates the probability of reaching a_D. Nevertheless, $P_D(H_i)$, for $i = 1, \ldots, 37$, is maximal (between 30 and 40%) when $w_1 > w_2$, that is, when NFκB-IκB pathway interactions are more frequent than caspase cascade interactions. (Figure 5d). In fact, it is interesting to note that, as the frequency of NFκB-IκB inter-

Weights	Survival probability	
(w_1, w_2, w_3)	$P_L(H_{38})$	$\frac{1}{37}\sum_{i=1}^{37} P_L(H_i)$
(1,1,1)	0.33	0.82
(1,1,5)	0.38	0.79
(1,5,1)	0.40	0.94
(1,5,5)	0.48	0.92
(5,1,1)	0.12	0.67
(5,1,5)	0.19	0.60
(5,5,1)	0.19	0.85
(3,1,1)	0.18	0.71
(5,1,1)	0.12	0.67
(10,1,1)	0.07	0.63

Table 4: Survival probability for different weight combinations.

actions becomes larger, also the probability of apoptosis increases (compare cases $w_1 = 1, 3, 5, 10$ with $w_2 = w_3 = 1$). In contrast, for the cases with $w_1 = 1$ (the NFκB-IκB interactions happen with smaller frequency), the probability of survival is significantly higher (larger than 80%) (Figure 5a-c). This suggests that the decision between cell survival and apoptosis strongly depends on the dynamics of the anti-apoptotic NFκB-IκB pathway: high "turnover rates" (that is, the interactions in this pathway happen at a faster rate) increase the probability of apoptosis.

The contribution of this Markov chain approach to the analysis of biological networks is thus two-fold: first, the common traits of the curves obtained with different matrices $\mathcal{P}^{scc}$ suggest global qualitative dynamical properties, which are robust with respect to the structure of the system, that is to say, independent of the choice of the updating order of the variables. Second, the quantitative aspects capture the variability and possible operating range of the network. Further applications of this technique include hypothesis testing and validation. For instance, one can easily analyze the effect of new interactions in the system's structure; similarly, the relative impact of two proteins can be studied by comparing the response of the system with different priority classes and updating strategies.

6. Conclusion

A method for model reduction of Boolean networks has been developed, based on the hierarchical decomposition of asynchronous graphs. The first aspect to be analyzed is the decomposition of the state space of the n-dimensional Boolean network into strongly connected components (SCCs), and the construction of the graph of transitions among them. The SCCs can be viewed as the "new states" of a "new" reduced system, since very often the number of SCC is less than or equal to the number of states of the original system. The second aspect in the model reduction method is the reconstruction of the Boolean rules that represent the graph of transitions among the SCCs. An identification algorithm (known as REVEAL) was adapted and used to determine a family of Boolean rules that describe the dynamics represented by a (sub-)graph of transitions.

More generally, the model reduction method uses the structure of interactions to isolate and identify smaller subsystems (or groups of variables and interactions) responsible for a given qualitative dynamical behaviour. This is a particularly relevant characterization for biological systems where experimental data consists (mostly) of qualitative measurements. The techniques described here are based on the fact that, for a Boolean system, the whole state space can, in principle, be enumerated; this introduces one other limitation to the method, on the size n of the network that can be computationally managed. Networks of intermediate size (up to $n = 15$) are easily computed. For larger networks, one may still use this method, by first isolating more basic modules. Each module would then be separately reduced and treated as one "node" with its Boolean rule.

As illustrated with the apoptosis example, model reduction using the asynchronous graph decomposition is a powerful potential source of valuable knowledge on a system. All the possible qualitative trajectories of the system are characterized, as well as their robustness to environmental perturbations. It is possible to identify the mechanism (in the form of smaller groups of variables and interactions) which is responsible for a given asymptotic behaviour of the system, for instance, the existence of oscillatory dynamics or (multi-)stability. Finally, the asynchronous transition graph can be naturally associated with a matrix of transition probabilities. Biological knowledge on the system's kinetics can thus be incorporated to obtain a more quantitative description of the system. These are also useful tools to test hypothesis and generate predictions concerning the structure of interconnections and the importance of each variable to the overall dynamics.

References

[1] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics, AMS Publications*, 280:193–219, 2001.

[2] A. Berman and R.J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Classics in Applied Mathematics. SIAM Press, 1994.

[3] R. Casey, H. de Jong, and J.L. Gouzé. Piecewise linear models of genetic regulatory networks: equilibria and their stability. *J. Math. Biol.*, 52:27–56, 2006.

[4] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA'03)*, volume 294 of *LNCIS*, pages 119–126. Springer, 2003.

[5] M. Chaves, R. Albert, and E.D. Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *J. Theor. Biol.*, 235:431–449, 2005.

[6] M. Chaves, T. Eissing, and F. Allgöwer. Regulation of apoptosis via the NFκB pathway: modeling and analysis. In A. Deutsch, N. Ganguly, and A. Mukherjee, editors, *Dynamics on and of complex networks*. Birkhauser, 2008. to appear.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2001. (2nd edition).

[8] N.N. Danial and S.J. Korsmeyer. Cell death: critical control points. *Cell*, 116:205–216, 2004.

[9] T. Eissing, H. Conzelmann, E.D. Gilles, F. Allgöwer, E. Bullinger, and P. Scheurich. Bistability analysis of a caspase activation model for receptor-induced apoptosis. *J. Biol. Chem.*, 279:36892–36897, 2004.

[10] B. Faryabi, J.-F. Chamberland, G. Vahedi, A. Datta, and E.R. Dougherty. Optimal intervention in asynchronous genetic regulatory networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):412–423, 2008.

[11] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.

[12] C. Frelin, V. Imbert, V. Bottero, N. Gonthier, A.K. Samraj, K. Schulze-Osthoff, P. Auberger, G. Courtois, and J. F. Peyron. Inhibition of the NF-κB survival pathway via caspase-dependent cleavage of the IKK complex scaffold protein and NF-κB essential modulator NEMO. *Cell Death Differ.*, 15:152–160, 2008.

[13] A.G. Gonzalez, A. Naldi, L. Sànchez, D.Thieffry, and C. Chaouiya. GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *BioSystems*, 84(2):91–100, 2006.

[14] J.-L. Gouzé. Positive and negative circuits in dynamical systems. *J. Biol. Sys.*, 6:11–15, 1998.

[15] C.M. Grinstead and J.L. Snell. *Introduction to probability*. AMS Bookstore, 1997.

[16] A. Hoffmann, A. Levchenko, M.L. Scott, and D. Baltimore. The IkB-NFkB signaling module: temporal control and selective gene activation. *Science*, 298:1241–1245, 2002.

[17] S.A. Kauffman. *The origins of order*. Oxford University Press (New York), 1993.

[18] B.N. Kholodenko, A. Kiyatkin, F.J. Bruggeman, E. Sontag, H.V. Westerhoff, and J.B. Hoek. Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc. Natl. Acad. Sci USA*, 99:12841–12846, 2002.

[19] S. Klamt, J. Saez-Rodriguez, J. Lindquist, L. Simeoni, and E. Gilles. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1):56, 2006.

[20] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architecture. In *Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[21] T. Lipniacki, P. Paszek, A.R. Brasier, B. Luxon, and M. Kimmel. Mathematical model of NFkB regulatory module. *J. Theor. Biol.*, 228:195–215, 2004.

[22] N.D. Perkins. Integrating cell-signalling pathways with NF-κB and IKK function. *Nature Rev. Mol. Cell Biol.*, 8:49–62, 2007.

[23] M. Rehm, H. Düßmann, R.U. Jänicke, J.M. Tavaré, D. Kögel, and J.H.M. Prehn. Single-cell fluorescence resonance energy transfer analysis demonstrates that caspase activation during apoptosis is a rapid process. *J. Biol. Chem.*, 277:24506–24514, 2002.

[24] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework. *Advances in Applied Mathematics*, 41(3), 2008.

[25] D. Ropers, H. de Jong, M. Page, D. Schneider, and J. Geiselmann. Qualitative simulation of the carbon starvation response in *Escherichia coli*. *Biosystems*, 84(2):124–152, 2006.

[26] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, and Burkhart Schraven. A logical model provides insights into t cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, Aug 2007.

[27] L. Sánchez and D. Thieffry. A logical analysis of the *Drosophila* gap-gene system. *J. Theor. Biol.*, 211:115–141, 2001.

[28] M. Schliemann. Modelling and experimental validation of TNFα induced pro- and antiapoptotic signalling. Master's thesis, University of Stuttgart, Germany, 2006.

[29] H. Schmidt and E.W. Jacobsen. Identifying feedback mechanisms behind complex cell behavior. *IEEE Control Syst. Mag*, 4:91–102, 2004.

[30] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *BioInformatics*, 18(2):261–274, 2002.

[31] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391:258–275, 2008.

[32] C. Soulé. Graphic requirements for multistationarity. *ComPlexUs*, 1:123–133, 2003.

[33] R. Thomas and R. D'Ari. *Biological feedback*. CRC Press, 1990.

[34] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–195, 2001.

[35] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11(1):170–179, 2001.

[36] L. Tournier. *Etude et modélisation mathématique de réseaux de régulation génétique et métabolique*. PhD thesis, LJK-IMAG (`http://tel.archives-ouvertes.fr/tel-00308533`), 2005.

[37] A. Wuensche. Basins of attraction in network dynamics: a conceptual framework for biomolecular networks. In G. Schlosser and G.P. Wagner, editors, *Modularity in development and evolution*, chapter 13, pages 288–311. Chicago University Press, 2002.